

REMARKS

The Examiner's comments have been carefully reviewed by the Applicant and amendments have been made to the specification and claims to overcome the formal objections and rejections raised by the Examiner.

So far as the specification is concerned, the Examiner correctly observes that the description of the **compare** flag on page 6, lines 11-14 is unclear. This paragraph has been amended to clarify this by expanding the description of the **compare** flag. In doing so, it is respectfully submitted that no new matter has been added, since the resulting description conforms to the description as filed. In this connection, the Examiner's attention is drawn to page 5, lines 1 to 5 and page 5, lines 18 to 28, which fully supports the requested clarification to page 6.

It is respectfully submitted that there is no ambiguity in claim 1 with regard to the term "running the program", whose meaning the Examiner questions in connection with the corresponding term employed in the Summary of the Invention on page 2, line 26. The program is indeed the same program in which the utility is embedded and is not intended to refer to the utility, which is invoked only on running the program. On running the program, the trace utility will also be invoked and in so doing will cause the trace to be read and therefore we believe that it is a correct definition to state that the program is run for reading the trace file, although it is acknowledged that the reading of the trace is a by-product of

the running the program and is actually carried out by the utility. Nevertheless, we have amended the claim so as to clarify this point since the Examiner is correct to observe that the *program* does not itself read the trace file; but when the utility is reached in the program code, the utility is invoked and reads the trace file. We believe that this is now more clearly reflected in the claim. Corresponding clarifications have likewise been made to claims 7, 18 to 21, 24 and 28 so as to overcome the rejection raised in para. 4 on page 4 of the Office Action.

The rejections raised by the Examiner under 25 U.S.C. § 112, first paragraph, (para. 2 on page 3) as containing subject matter which was not described in the specification in such a way as to enable one skilled in the art to make and/or use the invention is respectfully traversed. Specifically, there is no contradiction to what is stated in claims 1 and 2: rather the two claims define the invention from different perspectives.

Claim 1 defines the operations carried out when there already exists a trace file in which debugger commands have been previously embedded and the program is run so as to invoke the utility for reading the trace file and executing the debugger command. This implementation of the invention does not require any comparison of the content of the trace file with a program variable: rather it is simply assumed that the content of the trace file is not a program variable but

rather is a debugger command that may be executed by the debugger.

As against this, claim 2 defines the operations carried out even when no such trace file exists. In this case, the utility creates the trace file and writes the program variables to the file. In order to carry out debugger commands at one or more desired locations in the program, the program variables written to the trace file at each of the desired locations must first be replaced by respective debugger commands. A second pass is now required, whereby the program is re-run. On reaching each desired location the program variable is traced and will yield that value that was originally written to the trace file but was subsequently replaced by the debugger command. Thus, the comparison will indicate that the content of the trace file is different from the program variable: and so the content of the trace file is therefore construed as a debugger command that is invoked by the debugger.

By such means, two passes are required. During a first pass, the utility writes program variables to the trace file as is done by conventional Print statements that are inserted into program code for checking the values of program variables at corresponding locations of the program. Prior to running the program again, one or more of the values written to the trace file are replaced by corresponding debugger commands. This is not recited in claim 2 as a direct operation, because clearly it may be carried out by a different entity than the

person running the program and the intention of the claim is to afford protection primarily against the person who runs the program for invoking the utility rather than against the person who modifies the trace file, who would nevertheless be a contributory infringer if the intention were to use the modified trace file for invoking debugger commands according to the method of claim 2. Thus, during a second pass, the program is run again (after replacing one or more program variables in the trace file by debugger commands): and since the comparison indicates that the content of the trace file at one or more locations is different from the corresponding program variables, this allows the utility to construe that the content of the trace file is a debugger command, which can be implemented by the debugger.

Thus claim 2 does not *require* that program variables in the trace file be replaced or augmented: if they are not, then the debugger will simply not be invoked. But if any of the program variables in the trace file are replaced or augmented, then the debugger will be invoked as explained.

These comments should also address the Examiner's questions relating to claims 25 and 28 and also to claim 7. Specifically, in direct answer to his question on page 4, para. 4 of the Office Action: "Furthermore, it is stated in parent claims that the debugger commands are embedded in the trace file in place of current files. Is this how the commands are construed?" our answer is affirmative. That is to say, as explained above, it is the difference between a program

variable traced by the utility on running the program and reaching the desired location, and the corresponding location (or line) of the trace file, that permits the utility to determine that the corresponding location (or line) of the trace file is a debugger command.

And again in direct answer to his question in the same paragraph: "In step (a)(iii) a debugger command that was embedded in the trace file is executed. Where are the debugger commands embedded in the trace file?" we respond as noted above with regard to claim 2. The claims are intentionally silent as to who embeds the debugger commands into the trace file in place of program variables, since this need not necessarily be done by the person running the program and therefore to include this feature in the claim would constitute an unnecessary limitation.

Finally, the Examiner's suggestion that the claims are a "literal translation into English from a foreign document" is factually incorrect and his objection that they "are replete with grammatical and idiomatic errors" appears unwarranted. In fact, a native English speaker fully conversant with the English language drafted the application in English. But if nevertheless, the Examiner has found "grammatical and idiomatic errors", we ask him kindly to point these out and we will welcome the opportunity to correct them.


Accordingly, the objections and rejections are traversed and it is requested that all of the pending claims be allowed and that the application be found in allowable condition.

Appln. No. 09/630,411
Amdt. dated August 7, 2003
Reply to Office Action of May 7, 2003

Applicants respectfully request favorable
reconsideration and allowance.

Respectfully submitted,

BROWDY AND NEIMARK, P.L.L.C.
Attorneys for Applicant(s)

By 
Jay M. Finkelstein
Registration No. 21,082

JMF:ma
624 Ninth Street, N.W.
Washington, D.C. 20001
Telephone No.: (202) 628-5197
Facsimile No.: (202) 737-3528
G:\BN\C\cohn\Farchil\pto\Amendment-07AUG03.Doc